2006

Mac OS X Forensics

Philip Craiger
University of Central Florida, craigerj@erau.edu

Paul Burke
University of Central Florida

Scholarly Commons Citation
Craiger, P., & Burke, P. (2006). Mac OS X Forensics. *Advances in Digital Forensics II,* (). https://doi.org/10.1007/0-387-36891-4_13

Chapter 13

MAC OS X FORENSICS

Philip Craiger and Paul Burke

Abstract This paper describes procedures for conducting forensic examinations of Apple Macs running Mac OS X. The target disk mode is used to create a forensic duplicate of a Mac hard drive and preview it. Procedures are discussed for recovering evidence from allocated space, unallocated space, slack space and virtual memory. Furthermore, procedures are described for recovering trace evidence from Mac OS X default email, web browser and instant messaging applications, as well as evidence pertaining to commands executed from a terminal.

Keywords: Macintosh computers, Mac OS X forensics

1. Introduction

Since its introduction in 1984, the Apple Macintosh has an enjoyed a small, albeit vocal, user base. Nevertheless, it is surprising that very little has been published regarding forensic examinations of Macintosh computers.

This paper describes procedures for conducting forensic examinations of Apple Macs running Mac OS X. Due to space limitations, certain assumptions are made to limit the scope of our coverage. These assumptions are: (i) The forensic computer and the suspect's computer run version 10.4.3 of Mac OS X, the latest version as of November 2005; (ii) the suspect has not set the Open Firmware password (Open Firmware is a processor and system-independent boot firmware used by PowerPC-based Macs, analogous to the x86 PC BIOS); (iii) the suspect has not used encryption via the Mac OS X FileVault, a virtual volume encrypted with 128-bit AES; and (iv) the suspect's hard drive is formatted with the Hierarchical File System Plus, commonly referred to as HFS+, the default file system since Mac OS X's release in 2000.

2. Mac OS X Background

Mac OS X is the successor to the original Apple Macintosh operating system that debuted in 1984. It has an entirely different code base from the original Mac OS, and is partially based on the NeXTSTEP operating system code base. Mac OS X is a UNIX-based operating system that comprises a FreeBSD-based subsystem and a Mach 3.0 microkernel. Although Mac OS X has a tightly integrated user interface that is very "Apple-like," the underlying architecture is UNIX, with all the services and a command line interface that constitute the heart of UNIX operating systems.

3. Forensic Examination Procedures

Mac OS X provides a novel method for creating a forensic duplicate that requires placing the suspect's computer into *target disk mode*. This mode allows an examiner to create a forensic duplicate of the suspect's hard drive using a FireWire cable connecting the two computers. Target disk mode works with any version of Mac OS X or OS 8/OS 9 (predecessors to Mac OS X) with FireWire software version 2.3.3 or later [1].

Additionally, target disk mode supports an onsite preview of the contents of the suspect's hard drive(s). Onsite previews are used when law enforcement is interested in seizing a computer only if evidence of probative value exists on the hard drive. An onsite preview allows an agent to conduct a search at the scene to determine if evidence exists to warrant the seizure of the suspect's computer.

3.1 Creating a Forensic Duplicate

It is crucial that nothing causes the suspect's hard drive to mount in read/write mode. This is because the process of mounting in this mode can cause changes to numerous files on the hard drive. We have discovered that booting a Windows 98 system causes changes to more than 400 files; approximately 400 files are modified in response to a graceful shutdown as well. `Disk arbitration`, the service that controls automatic disk mounting, must be disabled on the forensic computer prior to connecting the forensic and suspect computers. Under Mac OS X this service is performed by the file:

```
/usr/sbin/diskarbitrationd
```

Disk arbitration can be disabled by: (i) moving the file from its directory and rebooting the forensic computer, or (ii) moving its preference file:

```
/etc/mach_init.d/diskarbitrationd.plist
```

to a different location and rebooting. After reboot, an examiner can check the forensic computer's system log, or list the running processes and search for the keyword "diskarbitrationd," to verify that disk arbitration is not running. Note that this method only works for Mac OS X 10.4 or later versions. Earlier versions require a different method for disabling disk arbitration. Readers are referred to [4] for a discussion of procedures for disabling automounting on systems running OS X versions earlier than 10.4.

The next step is to connect the forensic computer and the suspect's computer with a FireWire cable, and then power on the suspect's computer while holding down the "T" key on the suspect's computer keyboard. This brings the suspect's computer into target disk mode. If the operation is successful, a FireWire icon is displayed on the suspect's computer monitor.

```
$ sudo ioreg -c "IOMedia" | grep 'FireWire Target Media' -A 10
    | |   |           | |   |         +-o AAPL FireWire Target Media ...
    | |   |           | |   |           | {
    | |   |           | |   |           |   "Leaf" = No
    | |   |           | |   |           |   "Writable" = Yes
    | |   |           | |   |           |   "BSD Minor" = 3
    | |   |           | |   |           |   "Preferred Block Size" = 512
    | |   |           | |   |           |   "BSD Major" = 14
    | |   |           | |   |           |   "BSD Name" = "disk1"
    | |   |           | |   |           |   "Size" = 80026361856
    | |   |           | |   |           |   "Content Hint" = ""
    | |   |           | |   |           |   "Removable" = No
```

Figure 1. Device mapping information.

Next, it is necessary to identify the device file that maps to the suspect's hard drive: in UNIX-based systems, attached physical devices are represented via files located under the `/dev` directory. The Mac OS X utility `ioreg` is used to display the Input/Output registry for the Mac to determine which device files in the `/dev` directory are mapped to the suspect's hard drive. Within the output, it is necessary to search for the term "FireWire Target Media," as demonstrated in Figure 1.

Figure 1 indicates that the suspect's hard drive is mapped to `disk1`. The UNIX utility `dd` may now be used to create a forensic duplicate of the physical disk. The integrity of the duplicate can be verified using MD5 hashing:

```
$ dd if=/dev/disk1 of=./evidence.dd
$ md5 /dev/disk1 ./evidence.dd
MD5 (/dev/disk1)  = 1cd57cb5d2e438d5be4bca9cf1664986
MD5 (evidence.dd) = 1cd57cb5d2e438d5be4bca9cf1664986
```

The duplicate may now be imported into any forensic software that understands the raw `dd` format to begin a forensic examination.

3.2 Previewing a Hard Drive

Information about the suspect's hard drive is needed in order to preview its contents. Like other BSD-based systems, volumes (partitions) are represented by the nomenclature `/dev/disk{n}s{m}`, where `{n}` is a number that denotes the physical drive, and `s{m}` represents the slice number `{m}`. Slice is BSD nomenclature for a volume. This information can be determined using the `hdiutil` utility as shown in Figure 2.

```
$ sudo hdiutil pmap /dev/disk1
Partition List
## Dev_______ Type_______________ Name________ Start___ Size____ End____
 0 disk1s1    Apple_partition_map Apple              1       63       63
 1            Apple_Free                            64   262144   262207
 2 disk1s3    Apple_HFS           Apple_HFS     262208 105406472 105668679
 3            Apple_Free                     105668680   262144 105930823
 4 disk1s5    Apple_HFS           Apple_HFS  105930824 50370648 156301471
 5            Apple_Free                     156301472       16 156301487
```

Figure 2. Volume/partition information.

Figure 2 shows that the suspect's hard drive contains three volumes: `s1`, `s3` and `s5`. An onsite preview can be conducted by mounting one of the volumes as read-only and viewing the contents through a terminal. Before mounting the volumes, a directory (`evidence.s3`) is created on the forensic computer and the volume is manually mounted as follows:

```
$ sudo mount -t hfs -r /dev/disk1s3 evidence.s3/
```

Note that `-t hfs` specifies the type of file system (hierarchical file system, which is the Mac OS X default), and the `-r` flag indicates to mount the volume read-only. Files on the suspect's hard drive are viewed from the command line by changing to:

```
/Volumes/evidence.s3
```

on the forensic computer.

Files from the suspect's hard drive may be copied to the forensic computer without fear of causing any changes on the suspect's hard drive because it is mounted in read-only mode. We recommend using the command `cp -p` to copy files to the forensic computer because it maintains the original metadata (modified, accessed, changed timestamps, owner and group, permissions, etc.).

An onsite preview provides a view of files in allocated space only. The hard drive must be accessed at a physical level to access deleted files (in unallocated space) and slack space. Procedures for recovering evidence from unallocated space and slack space are described in the next section.

Target disk mode provides a very simple imaging and previewing solution for Macs. A second alternative not discussed here is the use of a bootable Linux CD (PowerPC version for pre-2006 Macs, or a x86 version as of January 2006) to create a forensic duplicate [6].

4. Recovering Deleted Files

A common first procedure performed during a forensic examination is to recover files from the Trash and deleted files; deleted files require access to the media at a physical level. Below we describe procedures for recovering deleted files in allocated, unallocated and slack space.

As with versions of Microsoft Windows, there are several methods to delete files under Mac OS X. From the desktop, a user can drag-and-drop a file onto the Trash icon. An alternative is to CTRL-Click the mouse over the file, which brings up a menu from which the *Move to Trash* option can be chosen. Both methods are analogous to dragging and dropping a file into the Recycle Bin in Microsoft Windows [9].

Similar to the behavior of the Windows Recycle Bin, files placed in the Trash are not deleted. Rather, the files are copied to a special hidden folder in the user's default directory, and deleted from their original locations. Thus, a copy of the file moved to the Trash exists in allocated space, and can be recovered by opening the Trash icon and moving the file from the Trash. Of course, the original deleted file resides in unallocated space.

The Trash is represented on the file system as a hidden folder, `.Trash`, under each user's home folder, as shown below:

```
~/.Trash pc$ ls -al
total 160
drwx------    5 pc  pc    170 Feb  4 11:49 .
drwxr-xr-x   47 pc  pc   1598 Feb  4 11:49 ..
-rw-r--r--    1 pc  pc  73028 Feb  3 16:40 Picture.1.png
-rwxr-xr-x    1 pc  pc    780 Feb  4 11:48 automount.sh
```

Note that the `.Trash` folder contains two deleted files that can be recovered by copying each file to the forensic computer. The modified, accessed and changed timestamps for the deleted files can be determined using the `stat -x` command as follows:

```
~/.Trash pc$ stat -x Picture.1.png
  File: "Picture.1.png"
  Size: 73028        FileType: Regular File
  Mode: (0644/-rw-r--r--)    Uid: ( 501/ pc)  Gid: ( 501/ pc)
Device: 14,2   Inode: 1454786    Links: 1
Access: Sat Feb  4 09:51:05 2006
Modify: Fri Feb  3 16:40:44 2006
Change: Fri Feb  3 16:40:44 2006
```

From a forensic standpoint it is important to note that dragging and dropping a file to the Trash does not change the file's date and time stamps. This enables an examiner to determine the original modified, accessed and changed timestamps for the file. However, it does not allow an examiner to determine when the file was placed in the Trash. In contrast, a file deleted from the command line is deleted in the traditional sense (explained below), bypassing the Trash.

When the Trash is emptied, the files are marked as deleted in the HFS+ catalog special file (analogous to the Master File Table in NTFS or the root directory in FAT-based file systems), and the blocks allocated in the allocation special file (analogous to the allocation bitmap file in NTFS or the FAT in FAT-based file systems [2]) are zeroed. However, the contents of the file remain until they are overwritten by the operating system.

An examiner must access the suspect's hard drive at a physical level to recover deleted files residing in unallocated space. A physical analysis views media without regard to the imposed file system. Deleted files can be recovered manually if the starting block and the size of the file are known. An examiner can identify a file's starting block by searching the media at a physical level, e.g., with a hex editor, and determining the beginning of the file by identifying specific keywords or a file signature. To demonstrate this, suppose that a deleted file's starting block is 4355500, and the file comprises 18 contiguous blocks. Given that the default block size under Mac OS X is eight sectors per block (512 bytes/sector × 8 = 4096 bytes/block), the deleted file can be recovered as follows:

```
$ dd if=/dev/disk1 of=./evidence bs=4096 skip=4355500 count=18
```

The block size is set to the default block size used by HFS+ (`bs=4096`) and 18 contiguous blocks (the number of blocks required to store the file) are recovered from the starting block of the file (`skip=4355500`) onwards. This procedure recovers all the data up to the end of the last sector of the block. Therefore, the procedure will also recover the slack space, i.e., unallocated space located after the end-of-file marker, up to the end of the last block of the file, unless the file size is a multiple of the block size, meaning there is no slack space. Note also that the procedure will recover the entire contents of a deleted file only if the file was not fragmented.

Foremost [12] is a command line utility that automates the recovery of evidence from unallocated space. It recovers evidence from `dd` images and several other image formats based on internal data structures or file signatures such as headers and footers. Foremost is open source, and distributed as source code that can be compiled under Mac OS X.

4.1 Recovering Evidence from Virtual Memory

Even when a file has been deleted and physically wiped using a secure deletion utility [5], there may be traces of the file in virtual memory. It is possible to recover the evidence by searching through virtual memory, which is represented as a file called `swapfile` located in the directory `/var/vm`.

```
$ ls -al /var/vm
total 131072
drwxr-xr-x    4 root  wheel       136 Oct 14 10:50 .
drwxr-xr-x   24 root  wheel       816 Oct 14 10:52 ..
drwx--x--x   18 root  wheel       612 Oct 11 11:20 app_profile
-rw------T    1 root  wheel  67108864 Oct 14 10:50 swapfile0
```

UNIX utilities can be used to search for keywords within `swapfile`. The command `strings -o` is executed on `swapfile` to extract human-readable content in the 7-bit ASCII range. Next, `grep` is used to search for an appropriate keyword as shown below.

```
$ sudo strings -o /var/vm/swapfile0 | grep cyberterror -C 2
34649683 Definitions of cybert#849CA.doc
34649751 PD+W8BNMSWD
34649786 cyberterrorism
34649847 KLittleBuddy:Users:pc:Desktop:cyberterrorism...
34650027 AUsers/pc/Desktop/cyberterrorism.doc
34650667 BNMSWD
34650707 craiger.pollitt.ch#FFFFFFFF.doc
```

Generally, this procedure is more successful at recovering textual information as opposed to binary or graphical data. Unfortunately, the BSD version of `strings` does not support searches for text encoded in UNICODE. However, Sleuth Kit [11] includes a utility called `srch_strings` that supports searches for text encoded in 16-bit or 32-bit UNICODE (bigendian or littleendian). Sleuth Kit is distributed as source code that can be compiled for Mac OS X. The `srch_strings` utility can be used to search `swapfile` for a UNICODE encoded keyword as demonstrated below.

```
/var/vm pc$ sudo srch_strings -e l swapfile0 | grep -in 'defense'
79:During the 1960s, the Department of Defense
```

5. Recovering Application-Related Evidence

Mac OS X includes an email reader (Apple Mail), a web browser (Safari), and an instant messaging application (iChat), all of which leave trace evidence on a hard drive. Knowing the location of this trace evidence and the format of the evidence are crucial during an onsite preview and when conducting a logical analysis in the laboratory. Below we describe trace evidence locations for each of these applications, as well as any special reformatting that must occur to make the evidence human readable. Additionally, we demonstrate the recovery of evidence from the UNIX command line, assuming that that the suspect used the FreeBSD subsystem in the commission of a crime.

5.1 Apple Mail

Apple Mail is a full-featured email application that supports multiple POP3 and IMAP accounts and advanced filtering. User email is stored in the directory:

```
/Users/<username>/Library/Mail
```

The `/Users` directory is the Mac analog of the UNIX `/home` directory, where user files are stored. Consistent with the UNIX philosophy for a multi-user system, each user has his/her own directory.

Apple Mail files were stored in `mbox` format prior to Mac OS X 10.4. The `mbox` files are simple flat text files with individual emails appended to the end of the file [10]. As of Mac OS 10.4, Apple changed the default format to `emlx`, where each email is in its own file in ASCII format. Apparently, the change from `mbox` to `emlx` was made to allow for more thorough indexing under Apple's Spotlight integrated search technology [3]. Because each email is a simple text file, UNIX utilities can be used to search for specific keywords within the files.

One alternative is to convert the `emlx` files to `mbox` format, and then import the `mbox` file into another (non Apple Mail) email application, e.g., Outlook, Thunderbird or Eudora. Unfortunately, it is not possible to directly import `emlx` files into a secondary mail application for viewing emails as these applications do not understand the `emlx` format.

5.2 Safari Web Browser

Apple's bundled web browser, Safari, has become the *de facto* standard browser for Mac users. Safari is gaining increasing market share as Internet Explorer is no longer bundled with the latest versions of Mac OS X.

A suspect's web browsing history, download history and bookmarks can be used as evidence in criminal cases. Under Mac OS X the files used to store this information are named `History.plist`, `Downloads.plist` and `Bookmarks.plist`, respectively, and are stored in:

```
/Users/<username>/Library/Safari
```

`History.plist` stores information about the web pages the user has visited along with recent web searches performed from Safari's Google search bar. `Bookmarks.plist` stores bookmarks and `Downloads.plist` stores the history of files downloaded from Safari. A forensic examiner can parse these files using the `defaults read` command. Below, the `History.plist` file is read to determine a suspect's web browsing history.

```
$ defaults read ~/Library/Safari/History
WebHistoryDates = (
"http://www.google.com/search?q=bomb+making&ie=UTF-8&oe=UTF-8";
lastVisitedDate = "153417328.9";
title = "bomb making - Google Search";
visitCount = 1;
---
"" = "http://slashdot.org/";
lastVisitedDate = "160765082.1";
title = "Slashdot: News for nerds, stuff that matters";
visitCount = 3;
```

Note that the full path to the file must be included (the `.plist` extension must be excluded) to parse the file properly. The snippet above indicates that the suspect performed a Google search for the keywords "bomb making," and visited Slashdot three times.

Safari cache files are located in:

```
/Users/<username>/Library/Caches/Safari
```

Below is a directory listing that displays the naming scheme used:

```
$ ~/Library/Caches/Safari/00/00 $ ls -al
total 256
drwx------   13 pc  pc    442 Aug 21 17:05 .
drwx------   18 pc  pc    612 Mar  6  2005 ..
-rw-------    1 pc  pc   1385 Jul 27 20:04 1113936647-3722688267.cache
-rw-------    1 pc  pc  48227 Aug 21 17:05 1199711745-0794204939.cache
```

The browser cache is separated into several files that span a set of two-digit numbered folders, each comprising another set of two-digit numbered folders. The files can be viewed by copying these folders to the forensic computer and opening them in Safari.

Safari cookie files are stored in the file:

```
/Users/<username>/Library/Cookies/Cookies.plist
```

As with the other `.plist` files, it is possible to transfer these files from the suspect's computer to the forensic computer and use Safari to view them by selecting *Preferences:Security:Show Cookies* from the menu.

5.3 iChat Instant Messaging

Instant Messaging (IM) applications are bundled with all major operating systems. Additionally, all major web-based mail providers, including AOL, MSN, Yahoo! and Gmail provide instant messaging services.

Different IM applications store trace evidence pertaining to instant message conversations differently. Some may store instant messages on servers only, whereas others may keep a history of previous conversations by default. iChat does not automatically store previous conversations; however, a preference allows users to store iChat conversations on the local machine. The default location for these stored iChats is:

```
/Users/<username>/Documents/iChats
```

Individual iChat sessions are named as follows:

```
<username> on <date> at <time>.ichat
```

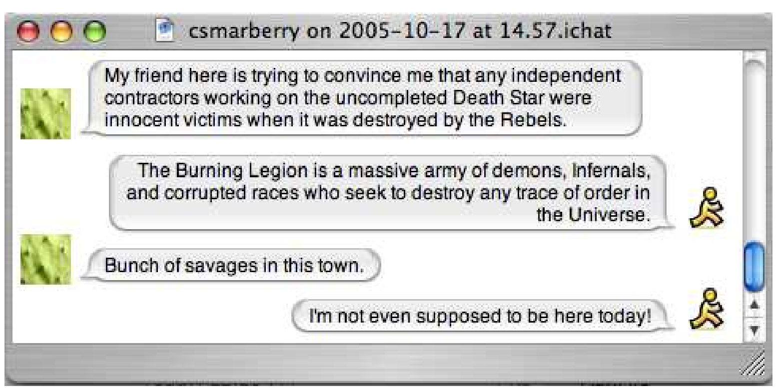

Figure 3. iChat

It is possible to view the contents of a session by copying a file to the forensic computer and opening the file by double clicking. Figure 3 shows a recovered iChat conversation between two users.

5.4 Command Line Input

A savvy criminal may open a terminal and use UNIX commands in the commission of a crime. This is most likely where an examiner will find evidence of a network intrusion.

The `bash` shell is the default shell for Mac OS X 10.4. It records commands executed by a user from the command line to a file named `.bash_history`, which is a hidden file in the user's home directory. To illustrate this, we copied the `.bash_history` file from the suspect's computer to the forensic computer. The last few entries in this file are:

```
$ tail .bash_history
...
sudo nmap -sS 192.168.1.0/24 > /Volumes/leet/recon.txt
cd /Volumes/leet/
less recon.txt
rm recon.txt
```

According to these entries, the suspect performed a port scan on an internal network (note the private addresses). The suspect saved the results to another volume named `leet`, viewed the file, and then deleted it. Unless this file has been overwritten, it can be recovered by accessing the `dd` image at the physical level, or accessing the associated `/dev` device in the target disk mode, and searching for keywords that we know existed in the deleted file:

```
$ cat ./evidence.dd | strings | grep -i 'nmap' -C 10
Starting nmap 3.81 ... at 2005-11-11 14:43 EST
Interesting ports on 192.168.1.2:
(The 1657 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
515/tcp  open  printer
548/tcp  open  afpovertcp
3689/tcp open  rendezvous
5000/tcp open  UPnP
```

Space limitations preclude us from discussing all the possible locations in a UNIX-based system from which evidence may be recovered. Interested readers are referred to [7, 8] for additional details.

6. Conclusions

Mac OS X forensics is an important but relatively unexplored area of research. This paper has discussed procedures for recovering evidence from allocated space, unallocated space, slack space and virtual memory, as well as Mac OS X default email, web browser and instant messaging applications, and command line input. Due to space limitations, it was not possible to cover other well-known applications, including email readers (e.g., Microsoft Entourage and Mozilla Thunderbird), web browsers (e.g., Microsoft Internet Explorer and Mozilla Firefox), and popular instant messaging applications (e.g., AOL AIM and Adium). It is important that forensic examiners become familiar with these applications and the locations of associated trace evidence. Examiners must also have a thorough understanding of the various versions of Mac OS X, some of which require specialized forensic procedures.

References

[1] Apple Computer, How to use FireWire target disk mode (docs.info.apple.com/article.html?artnum=58583), 2002.

[2] Apple Computer, Technical Note TN1150: HFS Plus Volume Format (developer.apple.com/technotes/tn/tn1150.html), 2004.

[3] Apple Computer, Working with Spotlight (developer.apple.com/macosx/spotlight.html), 2005.

[4] BlackBag Tech, FireWire target disk mode guidelines (blackbagtech.com/images/BBT_FireWire_Target_Mode.pdf), 2004.

[5] P. Burke and P. Craiger, Assessing trace evidence left by secure deletion programs, in *Advances in Digital Forensics II*, M. Olivier and S. Shenoi (Eds.), Springer, New York, pp. 185-195, 2006.

[6] P. Craiger, Recovering evidence from a Linux system, in *Advances in Digital Forensics*, M. Pollitt and S. Shenoi (Eds.), New York, pp. 233-244, 2005.

[7] D. Farmer and W. Venema, *Forensic Discovery*, Prentice-Hall, Upper Saddle River, New Jersey, 2004.

[8] K. Jones, R. Bejtlich and C. Rose, *Real Digital Forensics: Computer Security and Incident Response*, Addison-Wesley Professional, New York, 2005.

[9] Microsoft Corporation, How the recycle bin stores files (support.microsoft.com/default.aspx?scid=kb;en-us;136517), 2004.

[10] Network Working Group, RFC 4155 – The Application/Mbox Media Type (www.faqs.org/rfcs/rfc4155.html), 2005.

[11] Sleuthkit.org, Sleuth Kit (www.sleuthkit.org).

[12] Sourceforge.net, Foremost (foremost.sourceforge.net).